



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

DEFINICIÓN, DESARROLLO E IMPLANTACIÓN DE UN
SISTEMA DE GESTIÓN DEL CONOCIMIENTO Y SU
INTEGRACIÓN CON LA PLATAFORMA DE SOPORTE DE
SEGUNDO NIVEL (2LS) DE OPENBRAVO

Xabier Merino Zabaleta

Tutor: Jesús Villadangos

Pamplona, 1 Julio de 2011

ÍNDICE

1.- RESUMEN

2.- ¿QUÉ ES UN MÓDULO DE CONOCIMIENTOS?

2.1.- DEFINICIÓN

2.2.- ASPECTOS IMPORTANTES

2.3.- OBJETIVOS

2.4.- EL PASADO DE LAS BASES DE CONOCIMIENTO

2.5.- EL PRESENTE DE LAS BASES DE CONOCIMIENTO

3.- ¿QUÉ TENEMOS?

3.1.- SERVICIO DE SOPORTE EN OPENBRAVO

3.2.- ACTUAL BASE DE CONOCIMIENTO DE OPENBRAVO

4.- ¿QUÉ NECESITAMOS?

4.1.- NECESIDADES DEL EQUIPO DE SOPORTE

4.2.- CONCLUSIÓN

4.3.- ESTUDIO DE APLICACIONES

4.3.1.- MANTIS

4.3.2.- BUGZILLA

4.3.3.- OTRS

4.3.4.- REQUEST TRACKER

4.3.5.- FOOTPRINTS

5.- CONCLUSIÓN

5.1.- ELECCIÓN DE LA HERRAMIENTA

5.2.- BASE DE CONOCIMIENTO DE FOOTPRINTS

4.3.- LÍNEAS FUTURAS

6.- GLOSARIO

6.1.- ENTERPRISE RESOURCE PLANNING (ERP)

6.2.- POINT OF SALES

6.3.- SOPORTE TÉCNICO

6.4.- GESTOR DE INCIDENCIAS

6.5.- BUSSINES PARTNER

6.6.- ACUERDO DE NIVEL DE SERVICIO (SLA)

6.7.- WORKAROUND

6.8.- EQUIPO QA (QUALITY ASSURANCE)

6.9.- PARCHE

6.10.- VERSION O MAINTENANCE PACK

6.11.- BUG TRACKING

7.- BIBLIOGRAFIA Y REFERENCIAS

7.1.- BIBLIOGRAFÍA

7.2.- REFERENCIAS

1.- RESUMEN

El Proyecto Fin de Carrera que se presenta tiene como objeto la implantación de una Base de Conocimientos en el equipo de Soporte de la empresa Openbravo con el fin de ayudar a transferir el conocimiento y la experiencia de los diferentes miembros del equipo entre si.

Se empieza definiendo qué es una base de conocimientos, sus puntos importantes, los objetivos que se buscan con su implantación, su pasado y su futuro.

Tras esto se describe el servicio de Soporte que ofrece Openbravo, su modo de trabajo, sus características y parte de su actual base de conocimiento (*wikipedia* de Openbravo, *Forge* y repositorio de *issues*)

Se realiza un estudio dentro del propio equipo para definir una serie de características importantes de una base de conocimientos. Se decide estudiar aplicaciones que sean a la vez *Bug Tracking* y Base de conocimientos y se analizan 5 herramientas diferentes.

Se explica el porqué de la elección de una de estas herramientas y se pasa al análisis de la base de conocimientos que esta contiene. En este análisis se explican las opciones que ésta ofrece, desde la introducción de Soluciones hasta la posibilidad de valoración de estas, pasando por los diferentes tipos de búsqueda.

En el siguiente apartado se explica los siguientes pasos que se realizarán sobre el uso de esta base de conocimiento dentro del propio equipo de Soporte y en el resto de la empresa Openbravo.

Como último apartado tenemos el glosario. En él se describen términos utilizados durante la memoria (ERP, POS, soporte técnico, Gestor de incidencias o *Bug Tracking*, etc.)

2.- ¿QUÉ ES UN MÓDULO DE CONOCIMIENTO?

Un Módulo de Conocimiento (o *knowledge management* en inglés; KM) es un concepto aplicado en las organizaciones, que busca transferir el conocimiento y la experiencia existente entre sus miembros, de modo que pueda ser utilizado como un recurso disponible para otros en la organización.

En detalle, se refiere a las herramientas y a las técnicas diseñadas para preservar la disponibilidad de la información llevada a cabo por los individuos dominantes y facilitar la toma de decisiones, así como reducir el riesgo.

Usualmente el proceso implica técnicas para capturar, organizar, almacenar el conocimiento de los trabajadores, para transformarlo en un activo intelectual que proporcione beneficios y se pueda compartir.

2.2.- ASPECTOS IMPORTANTES

El más importante aspecto de una base de conocimiento es la calidad de la información que ésta contiene. Las Bases de Conocimiento de referencia tienen artículos cuidadosamente redactados que se mantienen actualizados al día, un excelente sistema de recuperación de información (Motor de Búsqueda), y un formato de contenido y estructura de clasificación bien definido.

2.3.- OBJETIVOS

El proceso de la Administración del Conocimiento, también conocido en sus fases de desarrollo como "aprendizaje corporativo" o "aprendizaje organizacional", tiene principalmente los siguientes objetivos:

- Identificar, recoger y organizar el conocimiento existente.
- Facilitar la creación de nuevo conocimiento.
- Apuntalar la innovación a través de la reutilización y del apoyo en la habilidad de las personas de la organización para lograr una mejor productividad en la empresa.

2.4.- EL PASADO DE LAS BASES DE CONOCIMIENTO

La transferencia del conocimiento (un aspecto de la Administración del Conocimiento) ha existido siempre como proceso en las organizaciones. De manera informal por medio de las discusiones, sesiones, reuniones de reflexión, etc., y de

2.5.- EL PRESENTE DE LAS BASES DE CONOCIMIENTO

Como práctica emergente de negocio, la administración del conocimiento ha considerado la introducción del principio oficial del conocimiento, y el establecimiento de *Intranets* corporativas, de *wikis*, y de otras prácticas de la tecnología del conocimiento y de la información.

Es un mercado del software y un área en la práctica de la consultoría, relacionada con disciplinas tales como la inteligencia competitiva. Un tema particular de la administración del conocimiento es que el conocimiento no se puede codificar fácilmente en forma digital, tal como la intuición de los individuos dominantes que viene con años de la experiencia y de poder reconocer los diversos patrones del comportamiento que alguien con menos experiencia no puede reconocer.

3.- ¿QUÉ TENEMOS?

3.1.- SERVICIO DE SOPORTE EN OPENBRAVO

La empresa Openbravo ofrece actualmente un servicio de soporte para los *business partners* o clientes finales con una Suscripción Profesional. Diferenciando estos dos grupos éste servicio es el siguiente:

1.- *Business Partners*:

- Mantenimiento, errores, actualizaciones y parches.
- Localización y resolución de problemas.
- Instalación de productos.
- Localización de productos.
- Resolución de incidencias técnicas.
- Realización y recuperación de copias de seguridad.
- Compatibilidad y rendimiento.
- Portal de soporte para la gestión de incidencias.

2.- Clientes finales:

- Actualizaciones periódicas y comprobaciones de seguridad.
- Copias de seguridad y procesos de recuperación para casos críticos.
- Soporte para el usuario fin en preguntas operacionales corrientes.
- Mejoras en el sistema según nuevos requisitos, tales como nuevos usuarios, informes o pequeñas personalizaciones.

Este servicio se ofrece a través de una aplicación *Helpdesk*. En el caso de Openbravo se utiliza la herramienta Footprints. Los clientes finales o *partners* introducen una incidencia en este portal y el equipo de soporte pasa a su resolución. Estas incidencias o *issues* pueden ser de tres tipos:

- Openbravo *Network Issues* (OBN *defects*): en este grupo se incluyen todas las incidencias relacionadas con bugs de la aplicación o productos de Openbravo.

- *Regular issues*: incluimos como regular issues todo tipo de dudas o preguntas relacionadas con la aplicación, soporte de consultoría, etc.
- *Extended Issues*: son extended issues aquellas incidencias relacionadas con las ayudas a desarrollos.

A su vez, estas incidencias son clasificadas según su severidad. En Openbravo se sigue la siguiente clasificación:

- *Critical*:
 - El sistema de producción está muy afectado o completamente caído.
 - Operaciones del sistema o aplicaciones críticas del negocio están caídas.
 - Todo el desarrollo se encuentra bloqueado.
- *Major*:
 - El sistema de producción funciona con capacidades limitadas.
 - El sistema de producción es inestable con interrupciones periódicas.
 - Un subsistema importante en el desarrollo está bloqueado.
- *Minor*:
 - Existen errores causando una parcial, no crítica, pérdida de funcionalidad. Afectan a algunas operaciones pero permite seguir funcionando.
 - Existe una falta de documentación o una necesidad de aclarar un procedimiento.
 - Hay errores en el desarrollo del sistema que pueden afectar al rendimiento.
- *Trivial*:
 - Existen errores en el desarrollo del sistema que tienen un pequeño impacto con el rendimiento.
- *Feature Request* (Mejora de la aplicación):
 - Existe una solicitud de una mejora del producto.

Existe un acuerdo de Nivel de Servicio (SLA) entre Openbravo y el *partner* o cliente final. Este acuerdo es un contrato escrito que se define con el fin de fijar el nivel acordado para la calidad del servicio. El SLA es una herramienta que ayuda a ambas partes a llegar a un consenso en términos de nivel de calidad de servicio, en aspectos

tales como el tiempo de respuesta, el tiempo de resolución, disponibilidad horaria, documentación disponible, personal asignado al servicio, etc.

Las siguientes tablas muestran los tiempos de respuesta y resolución definidos a nivel de contrato dependiendo del tipo de incidencia de que se trata y de la severidad de la misma.

– *Openbravo Network Issues:*

	Tiempo de Respuesta	Tiempo de Resolución
Critical	1 día	6 días
Major	3 días	6 semanas
Minor	1 semana	3 meses
Trivial	N/A	N/A
Feature Request	N/A	N/A

– *Regular issues:*

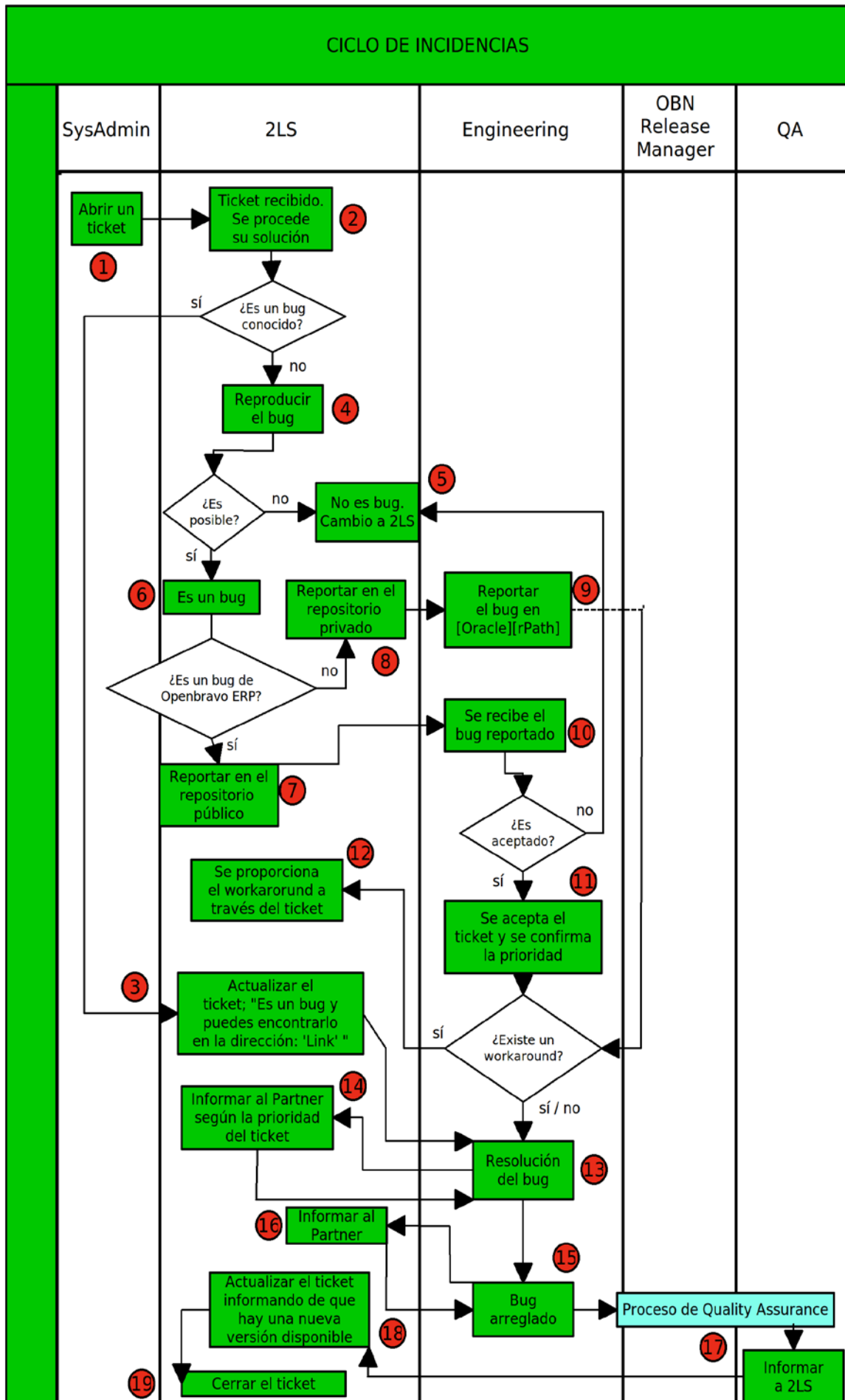
	Tiempo de Respuesta	Tiempo de Resolución
Critical	1 día	1 día
Major	1 día	3 días
Minor	1 día	5 días
Trivial	1 día	10 días

– *Extended issues:*

	Tiempo de Respuesta	Tiempo de Resolución
Critical	N/A	N/A
Major	2 días	5 días
Minor	2 días	10 días
Trivial	2 días	4 semanas

Los bugs de la aplicación o incidencias *Openbravo Network Issues* siguen un proceso más largo de resolución. Este proceso es más complejo y por ello conlleva un mayor tiempo de resolución ya que se trata de *bugs* de la aplicación. Estos *bugs* tienen que ser resueltos de la mejor manera posible, sin crear errores a consecuencia de su solución.

A su vez, estas soluciones tienen que ser testeadas por el equipo de QA (*Quality Assurance*) y empaquetadas a tiempo para su introducción en la *Maintenance Pack* correspondiente. A continuación un esquema y su explicación sobre el proceso completo de este tipo de incidencias.



Descripción de los pasos mostrados en el esquema anterior:

1.- Un partner abre un ticket en el portal de soporte para informar de un bug encontrado en Openbravo Network.

2.- El equipo de 2LS recibe el ticket e informa al partner de que se va a proceder a su resolución.

3.- Si el bug reportado es un bug conocido, el equipo de soporte informa al partner de que el equipo de *Engineering* está trabajando en él.

4.- Si el bug reportado es un bug desconocido, el equipo de soporte intenta reproducirlo en la última versión de la aplicación en la que ha sido reportado el bug.

5.- Si el bug no se puede reproducir se le informa al partner de que el bug será tratado como un bug de la aplicación por lo que se cambiará el ticket a 2LS con el correspondiente consumo de su crédito de soporte.

6.- Si el bug ha sido reproducido por el equipo de 2LS, se le informa al partner de que ha sido reproducido y que será reportado en el repositorio de bugs de Openbravo.

7.- El bug es reportado en el repositorio de bug, donde es clasificado y asignado según sus características al equipo de *Engineering*. Informar al partner de que el bug ha sido reportado añadiendo su dirección del repositorio.

8.- Errores de Oracle, de rPath y otros componentes son reportados en un repositorio privado.

9.- El administrador de la versión recibe los bugs que no pertenecen a Openbravo ERP y los reporta a los repositorios específicos.

10.- El equipo de *Engineering* recibe el bug reportado.

11.- El bug es aceptado por el equipo de *Engineering* y ellos confirman o cambian la prioridad.

12.- Si hay un proceso alternativo para usar mientras el bug es solucionado se le informa al partner y se le recomienda usarlo.

13.- El equipo de *Engineering* trabaja en la resolución del bug.

14.- Mientras el bug está siendo solucionado, el equipo de 2LS informa periódicamente al partner de las novedades que se produzcan. La periodicidad depende de la prioridad del bug.

15.- El bug es arreglado por el equipo de *Engineering*.

16.- Se informa al partner de que el bug está solucionado y que sera añadido en la siguiente versión de Openbravo.

17.- El equipo de QA informa al equipo de 2LS que hay un parche disponible para descargar de la consola de *Openbravo Network Management*.

18.- Se informa al partner de que el bug está solucionado y que se incluirá en la siguiente versión de Openbravo ERP. Cuando la última versión es publicada se le informa al partner.

19.- Se cierra el bug.

3.2.- ACTUAL BASE DE CONOCIMIENTO DE OPENBRAVO

Actualmente podemos diferenciar en Openbravo varios sectores que podríamos decir que forman parte de nuestra base de conocimientos actual. Este conocimiento público se basa en toda la documentación relacionada con la aplicación, toda la información sobre los módulos y proyectos de Openbravo (en este caso muchos de ellos tienen acceso limitado), bug y mejoras de la aplicación introducidas en el repositorio de incidencias (issues), etc.

1.- Wiki:



En

'<http://wiki.openbravo.com>' podemos encontrar en varios idiomas diferentes recursos y documentación sobre Openbravo ERP y Openbravo POS.

La documentación de Openbravo ERP consta de:

- Información General: cómo empezar con Openbravo ERP, requisitos del sistema, FAQ (preguntas más frecuentes), artículos generales, etc.
- Instalación: aplicaciones virtuales de Openbravo, instalación de Openbravo ERP, artículos de instalación, etc.
- Desarrollo: Openbravo ERP *roadmap*, guía del desarrollador, fragmentos de código, artículos de desarrollo, etc.
- Documentación del usuario: manual de Openbravo 2.50, documentación funcional, glosario de Openbravo ERP, artículos de usuario, etc.

- QA (garantía de calidad): directrices de bugs reportados, artículos de QA, etc.
- Arquitectura: conceptos de desarrollo principal de Openbravo ERP, *Data Access Layer*, modularidad, artículos de arquitectura, etc.
- Comunidad: guía del contribuidor, proyectos de comunidad (nuevos desarrollos de funcionalidad, ...), canales de comunicación (forums, weblogs, listas de e-mails, ...), artículos de comunidad, etc.
- Configuración y desarrollo: manual de configuración de Openbravo ERP, consejos de seguridad y rendimiento, artículos de configuración, etc.
- Otros recursos de Openbravo: documentación adicional de Openbravo, repositorio de bugs, recursos externos acerca de Openbravo, etc.

La documentación de Openbravo POS consta de:

- Información general de POS: especificaciones técnicas de Openbravo OPS, FAQ de Openbravo POS.
- Desarrollo: Openbravo POS roadmap, guía del desarrollador de Openbravo POS, proyectos y localización de Openbravo POS, fragmentos de código.
- Documentación del usuario: manual del usuario de Openbravo POS (español e inglés).
- QA (garantía de calidad): tests aceptados.
- Instalación y configuración: guía de instalación de Openbravo POS, guía de instalación en PDA de Openbravo POS, guía de instalación de Hardware de Openbravo POS, etc.
- Comunidad: guía del contribuidor de Openbravo POS, proyectos de comunidad de Openbravo POS.
- Configuración y desarrollo: guía del administrador de Openbravo POS, integración de Openbravo POS y Openbravo ERP.



2.- Mantis Bug Tracke. Repositorio de Issues.



MantisBT es un sistema gratuito de seguimiento de incidencias basado en Web. Está escrito en PHP y trabaja con MySQL, MS SQL y bases de datos PostgreSQL y webserver.

Desde aquí llevamos el seguimiento de todos los issues [1]. Centrándonos en el tema de los conocimientos en Openbravo, Mantis es el lugar donde se encuentran todos y cada uno de los issues reportados, por lo que adquiere un papel muy importante en la búsqueda de información.

Actualmente, en Openbravo, tenemos dos maneras de realizar la búsqueda de issues en Mantis. La primera es utilizando el filtro de búsqueda que contiene, éste es muy completo ya que contiene varios filtros para realizar una búsqueda más precisa. Podemos buscar los issues a través de varios tipos filtros:

- Personas relacionadas con el issue: quién lo a reportado o a quién ha sido reportado.
- Características del issue: categoría, tipo, prioridad, estado, etc.
- Características del sistema donde se ha reproducido el issue: plataforma, sistema operativo, versión del sistema operativo, etc.

La otra forma de búsqueda de issues en mantis es a través de un buscador personalizado de Google. Con esto realizamos una búsqueda tipo Google centralizada en la página issues.openbravo.com.



3.- Openbravo Forge:



Forge [2] es un portal que permite a la comunidad de Openbravo desarrollar, compartir y descargar soluciones de Openbravo. En la actualidad, la Forja incluye contribuciones realizadas por más de 8000 desarrolladores y unos 200 proyectos. Un repositorio central integrado en Openbravo Forge Proporciona diversos servicios y módulos de Openbravo.

4.- ¿QUÉ NECESITAMOS?

4.1.- NECESIDADES DEL EQUIPO DE SOPORTE

Para responder a esta pregunta es necesario conocer las características de una *Knowledge Base* que serían beneficiosas para el equipo de Soporte.

Las necesidades serían las siguientes:

1.- Fácilmente actualizable / facilidad de introducción de datos.

La introducción y modificación de incidencias son las acciones más repetidas al cabo de una jornada laboral por un miembro del equipo de Soporte, por lo que es muy importante que la introducción de soluciones en la *Knowledge Base* se puedan llevar a cabo de una manera rápida y sencilla, evitando así un gran aumento de la carga de trabajo.

2.- Eficacia y eficiencia de las búsquedas:

Una de las características más importantes para que una *Knowledge Base* sea de calidad es que sus búsquedas sean eficaces y eficientes. Entendemos como eficaz una búsqueda que encuentra lo que realmente se está buscando y como eficiente una búsqueda que no implique un gran coste temporal.

3.- Búsquedas personalizadas:

Partiendo del punto dos, un concepto que hay que tener en cuenta para ahorrar esfuerzo y tiempo en las búsquedas es el de búsquedas personalizadas. Con este tipo de búsquedas podremos guardar unas características de búsqueda ahorrándonos la acción de introducirlas cada vez que queramos realizar dicha búsqueda.

4.- Categorizado:

La clasificación por categorías de *issues* en un *bug tracking* o de soluciones en una *Knowledge Base* sin necesidad de utilizar búsquedas es un aspecto muy importante desde el punto de vista organizativo.

5.- Nueva entrada a través de un email:

En relación al punto uno, de cara a facilitar la introducción de nuevas incidencias o soluciones, la posibilidad de crear estas entradas a partir de e-mails aportaría una

mayor rapidez al proceso, al evitarnos de este modo la introducción manual de la misma.

6.- Posibilidad de obtener informes o estadísticas:

La obtención informes en una *Knowledge Base* aportaría la posibilidad de acceder a las soluciones de una manera general. Del mismo modo ayudaría a realizar estudios sobre la situación de la *Knowledge Base*. Estos informes serían útiles para cualquier tipo de consulta sobre un tema en concreto.

7.- Posibilidad de comentar o valorar una solución:

Con valoración de soluciones se destaca cual ha sido mas útil y cual menos, así podremos captar las soluciones menos útiles y modificarlas o eliminarlas. Un aspecto interesante sería la capacidad de comentar soluciones, de esta manera sería posible aumentar su calidad.

4.2.- CONCLUSIÓN

Buscamos la herramienta que aúne el mayor número de estas necesidades. Basándonos en la primera de ellas (Fácilmente actualizable) concluimos que sería necesario que la *Knowledge Base* estuviese integrada en un *Bug Tracking*. De esta manera agilizaríamos en gran manera el proceso de introducción de datos al tener, tanto el ticket de origen como la *Knowledge Base* de destino en la misma aplicación.

4.3.- ESTUDIO DE APLICACIONES

1.- MANTIS



MantisBT [3] es un software de código abierto para el control de errores basado en una arquitectura web. Está escrito en lenguaje PHP y trabaja con MySQL, MS SQL y base de datos PostgreSQL y un servidor web.

1.- Fácilmente actualizable / facilidad de introducción de datos.

Main	My View	View Issues	Report Issue	Change Log	Roadmap	Docs	Wiki	My Account	Billing	Logout
----------------------	-------------------------	-----------------------------	------------------------------	----------------------------	-------------------------	----------------------	----------------------	----------------------------	-------------------------	------------------------

La introducción de datos en Mantis se hace a través de la opción 'Report Issue' del menú principal de la aplicación. Esta opción nos llevará a la ventana de introducción de *issues*.

Enter Report Details	
*Category	(select) ▼
Reproducibility	have not tried ▼
Severity	minor ▼
Priority	normal ▼
Select Profile	▼
OR Fill In	
Platform	<input type="text"/>
OS	<input type="text"/>
OS Version	<input type="text"/>
Product Version	▼
Product Build	<input type="text"/>
Assign To	<input type="text"/>
Target Version	▼

Existen dos partes en la introducción de issues en Mantis. La primera se centra en la introducción de diferentes datos referentes al issue. Desde la categoría hasta la persona a la que se le va a asignar el issue, pasando por la versión del producto en la que se produce o la prioridad y criticidad del mismo.

La segunda parte se centra en el contenido del issue. Esta parte consta de título del *issue*, su descripción, los pasos para reproducirlo e información adicional.

Podemos concluir que la introducción de datos se puede realizar de manera sencilla, ya que, a pesar de existir muchos campos, no son obligatorios, por lo que rellenando los que realmente tienen importancia podemos introducir una incidencia de manera rápida y a la vez de calidad.

2.- Eficacia y eficiencia de las búsquedas.

Existen varias posibilidades de búsqueda en Mantis. Entre ellas podemos destacar dos:

1.- La búsqueda rápida a través del número de issue.

Esta acción se lleva a cabo a través del pequeño buscador situado en la parte superior derecha de la pantalla. Introduciendo el número del issue accedemos de manera rápida a la incidencia deseada.



2.- Búsqueda avanzada.

Reporter:	Monitored By:	Assigned To:	Category:	Severity:	Resolution:	Profile:
any	any	any	any	any	any	any
Status:	Hide Status:	Product Build:	Product Version:	Fixed in Version:	Priority:	Target Version:
any	closed ▾	any	any	any	any	any
Show:	View Status:	Show Sticky Issues:	Changed(hrs):	Use Date Filters:	Relationships:	
50	any	Yes	6	No	any	
Platform:	OS:	OS Version:	Tags:			
any	any	any				

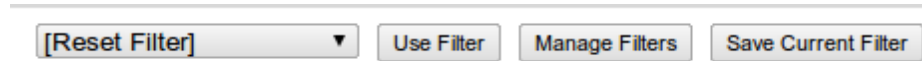
Accedemos mediante la opción '*View Issues*' del menú principal. El principal problema de esta búsqueda es que la introducción de datos es muy pesada ya que para la elección de cada característica es necesario abrir un desplegable. Teniendo en cuenta que existen varios campos que se pueden completar es una acción bastante costosa para el usuario.

Centrándonos en los resultados de las búsquedas, desde nuestra experiencia (ya que utilizamos esta aplicación para llevar el control de bugs de la aplicación) podemos afirmar que para nada es eficaz. En la inmensa mayoría de los casos no somos capaces de encontrar lo que buscamos.

Por todo esto, Mantis no tiene un sistema de búsqueda eficaz.

3.- Búsquedas personalizadas.

Mantis tiene la posibilidad de crear búsquedas personalizadas. Para ello bastaría con rellenar los términos deseados para una búsqueda avanzada y guardar esta personalización a través de las opciones situadas en la parte inferior derecha del cuadro de características.



4.-

Categorizado.



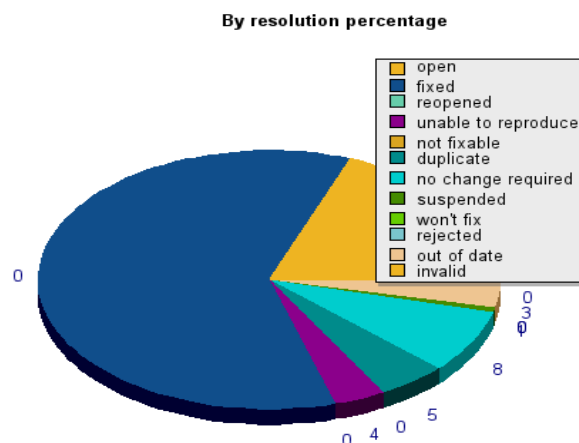
A través de la opción 'My View' del menú principal accedemos a una vista de todos los issues clasificados por sus estados (*Unassigned, Reported by Me, Resolved, Recently Modified*)

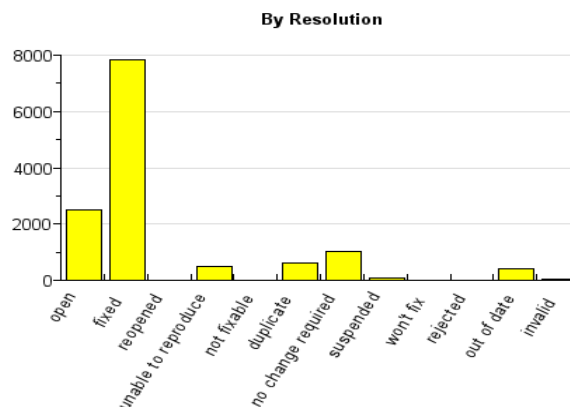
5.- Nueva entrada a través de un e-mail.

En el caso de Mantis no existe la posibilidad de añadir una nueva solución o incidencia a través del e-mail.

6.- Posibilidad de obtener informes o estadísticas.

Existe la pestaña 'Summary', en la que se ofrece la vista de varias estadísticas de las incidencias. Estos datos se dividen dependiendo de su estado, su prioridad, su severidad, su categoría y su resolución. Mantis permite imprimir estas estadísticas. Estos son algunos ejemplos:





7.- Posibilidad de comentar o valorar una solución.

Se ofrece la posibilidad de añadir dos tipos de nota a cada issue reportado en Mantis, pública o privada. Lo que no es posible es valorar ninguna de estas incidencias.

Add Note

Note

View Status

☐ private

Time tracking

00:00
Start
R

Add Note

2.- BUGZILLA



Bugzilla [4] es una herramienta basada en web de seguimiento de errores, originalmente desarrollada y usada por el proyecto Mozilla. Utiliza un servidor HTTP y una Base de Datos (MySQL).

1.- Fácilmente actualizable/introducción de datos.

La introducción de issues en Bugzilla se divide en tres grandes pasos:

1.1.- ¿Tu bug ya ha sido reportado?

En este apartado se muestra una tabla con los últimos bug reportados. Esta tabla viene con un pequeño buscador.

1.2.- Introducción de la información del issue:

En este apartado reside la información importante del issue y por consiguiente el paso al que más tiempo tendremos que dedicar. Los campos a rellenar son los siguientes:

- Hardware Platform
- Operating system
- Build Identifier: se introducirá la versión exacta del producto usado.
- URL: dirección en la que se muestra el bug.
- Summary: frase que resuma el bug.
- Details: descripción mas profunda del bug
- Reproducibility.
- Steps to reproduce: pasos a seguir para reproducir el problema.
- Actual results: descripción del resultado que se consigue al seguir los pasos anteriores.
- Expected Results: descripción del resultado que se esperaba conseguir.
- Additional Information.
- Severity: Trivial, Minor, Normal, Major y Critical.

1.3.- Reportar el bug:

Finalmente se pulsará el botón pertinente para concluir la introducción del *issue*.

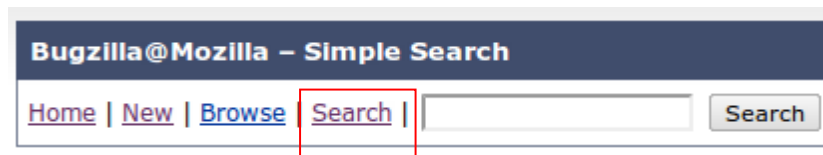
Al igual que sucedía con Mantis existen muchos campos para rellenar, pero la posibilidad de dejarlos en blanco hace que esta introducción se pueda calificar como fácil y rápida.

2.- Eficacia en búsquedas.

En este caso también se divide la búsqueda ofrecida por Bugzilla en simple y avanzada:

2.1.- Simple:

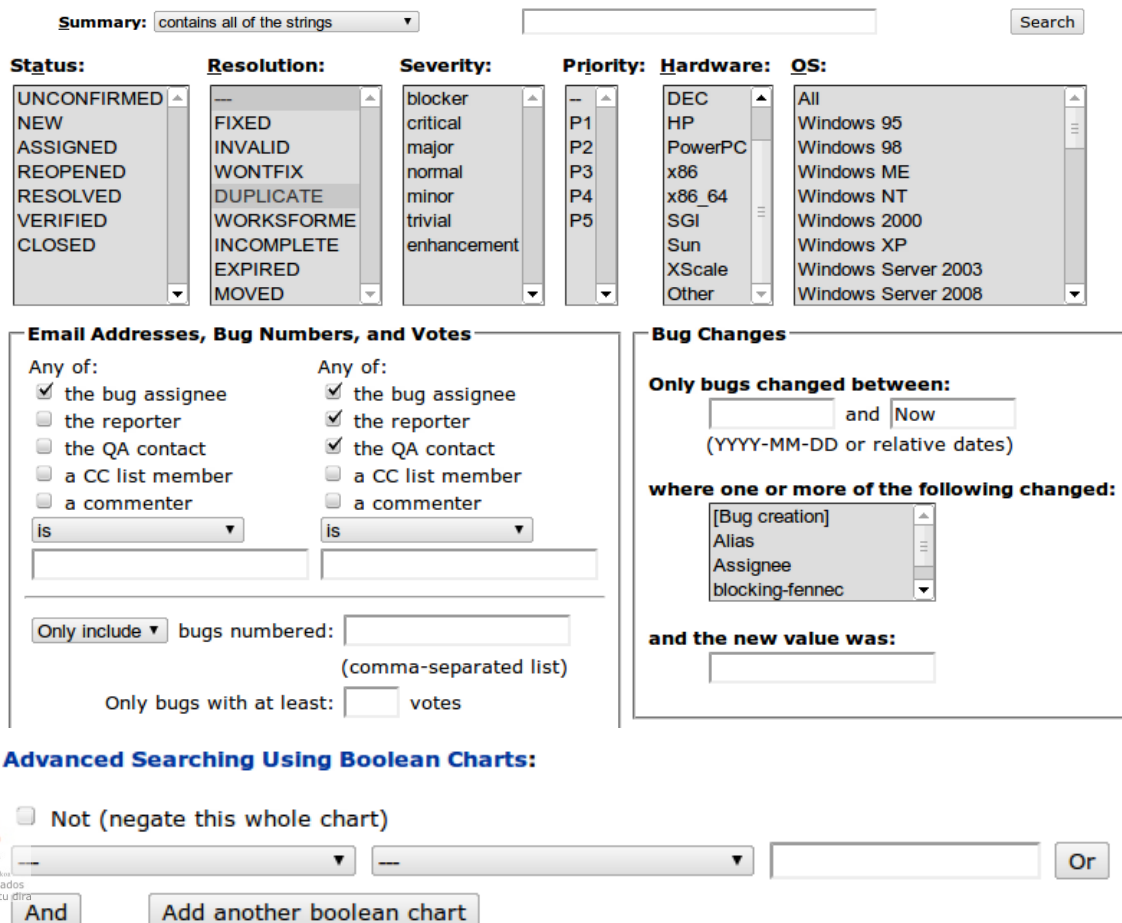
Existe la posibilidad de realizar una búsqueda simple a partir de la opción ofrecida en el menú principal.



En la información del BugTracker podemos comprobar que esta búsqueda simple gana en funcionalidad con la posibilidad de la utilización de palabras clave. Por ejemplo, para realizar una búsqueda rápida de *issues* con estado nuevo bastaría con introducir '*stat:NEW*' o para buscar los bugs asignados a una persona determinada introduciríamos '*assigned_to:Xabier*'

2.2.- Avanzada:

La búsqueda avanzada de Bugzilla es del mismo tipo que la de Mantis, muchos campos a rellenar.

The image displays the 'Advanced Search' form in Bugzilla. At the top, there's a 'Summary' dropdown set to 'contains all of the strings' and a search input field with a 'Search' button. Below this are several filter sections: 'Status' (listing UNCONFIRMED, NEW, ASSIGNED, REOPENED, RESOLVED, VERIFIED, CLOSED), 'Resolution' (listing FIXED, INVALID, WONTFIX, DUPLICATE, WORKSFORME, INCOMPLETE, EXPIRED, MOVED), 'Severity' (listing blocker, critical, major, normal, minor, trivial, enhancement), 'Priority' (listing P1, P2, P3, P4, P5), 'Hardware' (listing DEC, HP, PowerPC, x86, x86_64, SGI, Sun, XScale, Other), and 'OS' (listing Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008). The 'Email Addresses, Bug Numbers, and Votes' section allows filtering by assignee, reporter, QA contact, CC list member, or commenter, with options to include or exclude bugs numbered in a comma-separated list or by a minimum number of votes. The 'Bug Changes' section allows filtering by date range and by specific fields that changed (Bug creation, Alias, Assignee, blocking-fennec) and the new value. At the bottom, the 'Advanced Searching Using Boolean Charts' section includes a 'Not (negate this whole chart)' checkbox, a series of input fields for boolean logic, and buttons for 'And', 'Add another boolean chart', and 'Or'.

El sistema de búsqueda de Bugzilla mejora ligeramente al de Mantis al permitir realizar búsquedas simples con palabras clave. Pero, al igual que pasaba en la anterior herramienta, no es eficaz a la hora de realizar búsquedas sobre el campo *Summary* (búsqueda más utilizada).

3.- Búsquedas personalizadas.

En este caso Bugzilla no permite la opción de guardar búsquedas predefinidas. Este dato unido a la descripción anterior hace que las búsquedas sean uno de los puntos débiles de esta herramienta.

4.- Categorizado.

La categorización de los *issues* en Bugzilla es mínimo. El menú principal solamente ofrece la opción de ir a '*My Bugs*'.

Este es un aspecto importante que para nada cubre esta herramienta.

5.- Nueva entrada a través de un email.

Tampoco permite la introducción de *issues* a través del e-mail.

6.- Posibilidad de obtener informes o estadísticas.

Desde la opción '*Reports*' del menú principal accedemos a la opción de los informes que ofrece Bugzilla.

[\[?\]](#) | [Reports](#) | [My Requests](#) | [Preferences](#) | [Help](#) | [Log out](#)

Current State

- [Search](#) - list sets of bugs.
- [Tabular reports](#) - tables of bug counts in 1, 2 or 3 dimensions, as HTML or CSV.
- [Graphical reports](#) - line graphs, bar and pie charts.

- Search: Esta opción te lleva directamente a la pestaña 'Simple Search' de las búsquedas.
- Tabular Reports:

En esta opción observamos que, aparte de los menús que hemos visto en las búsquedas avanzadas, aparecen varias opciones para definir nuestros informes.

Vertical Axis:
 Classification

Horizontal Axis:
 Product

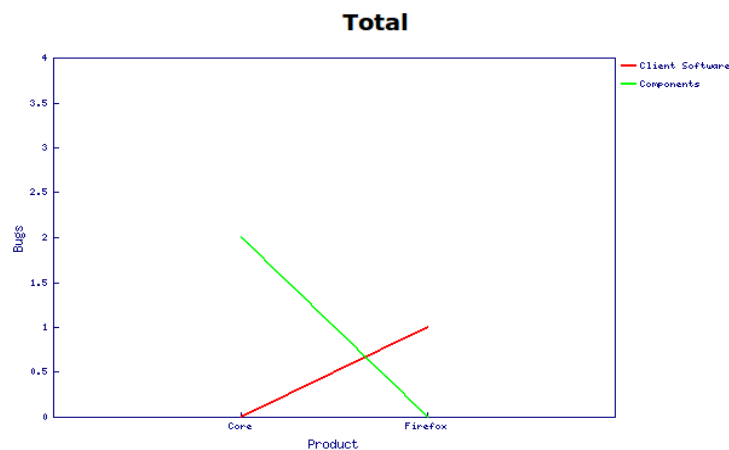
Multiple Tables:
 Component

Una vez hayamos definido cómo y qué mostrar en el informe, la herramienta nos muestra los datos del informe en formato 'table':

Total

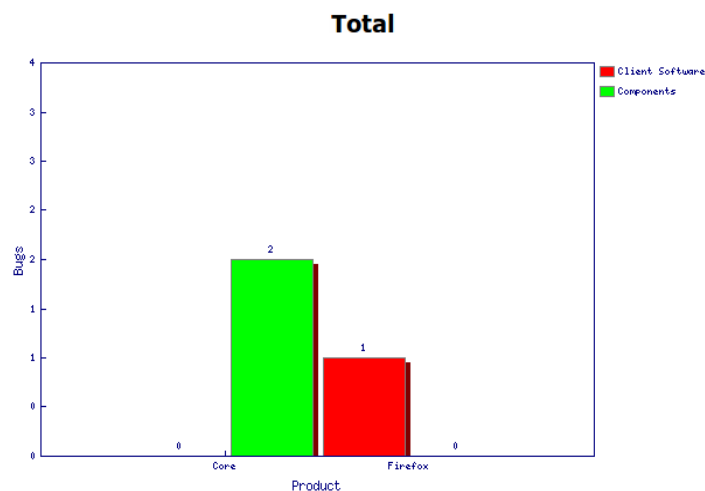
	Product		
	Core	Firefox	Total
Classification	Client Software	1	1
	Components	2	2
	Total	2	3

Existe la opción de mostrar los datos en gráficas lineales:



En gráficos de barra:

Y, en el caso de que sea necesario, la posibilidad de exportar en formato CSV. En definitiva, una funcionalidad muy completa en lo que a informes y estadísticas se refiere.



7.- Posibilidad de comentar o valorar una solución.

Si que existe la posibilidad de realizar comentarios, pero no de valorar el *issue* o solución.

3.- OTRS

((otrs))

Son las siglas en inglés de Open-source Ticket Request System. OTRS [5] es un sistema de software de código abierto para la gestión de procesos de negocio, Help Desk para soporte.

1.- Fácilmente actualizable/introducción de datos.



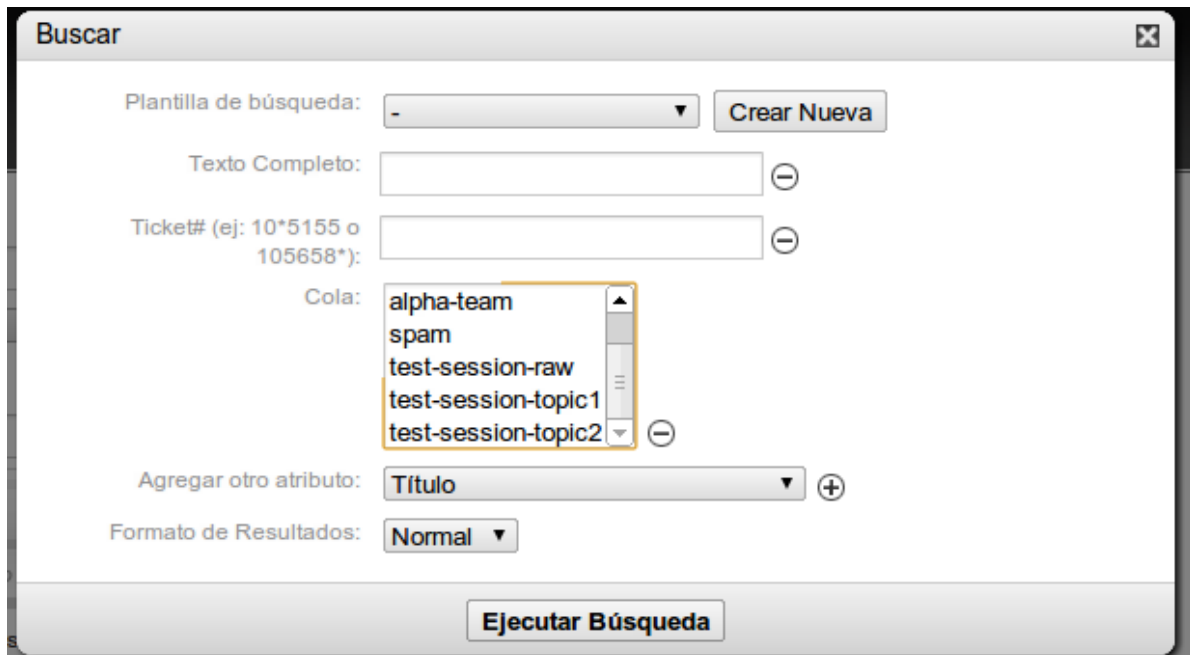
Se accede a la ventana de introducción de issues a través del menú principal, seleccionando la opción 'New Ticket'. Los campos a rellenar son los siguientes:

- From Customer: es un campo obligatorio en el que debes añadir la dirección de correo del cliente en cuestión. Si no es un e-mail correcto salta un error.
- To Queue: eliges el grupo al que quieres añadir el ticket. Obligatorio.
- Asunto.
- Texto.
- Adjuntar Archivo.
- Nº Cliente.
- Nuevo estado del ticket.
- Prioridad.
- Unidades de tiempo.

El número de campos a rellenar es menor y la mitad no son obligatorios. Estos campos son suficientes para aportar la información necesaria en el tratamiento del *issue*, por lo que me parece una forma ágil y sencilla de añadir nuevas incidencias.

2.- Eficacia en búsquedas.

Las búsquedas se realizan a partir de una ventana única. En este caso no existe una búsqueda simple y una búsqueda avanzada.



Como se ve en el primer campo existe la posibilidad de crear plantillas de búsquedas predeterminadas. Esta ventana de búsqueda es dinámica, puedes ir añadiendo campos a rellenar a través de '*Agregar Otro Atributo*'. Este factor influye en el tiempo de uso de las búsquedas, ya que de esta manera evitas buscar el campo que quieres rellenar, simplemente lo seleccionas. Este punto hace que podamos calificar las búsquedas en OTRS de eficaces.

3.- Búsquedas personalizadas.

Como se comenta en el apartado anterior es posible crear plantillas de búsquedas predefinidas.

4.- Categorizado.

La categorización es muy amplia. Nada más acceder a la herramienta aparece el panel principal. Este panel es configurable, pero si seleccionamos las opciones disponibles este panel divide los tickets en 4 grupos:

- Tickets de recordatorios.
- Tickets escalados.
- Nuevos tickets.
- Tickets abiertos.

Esta opción es muy útil, dado que de un vistazo puedes comprobar el estado de los tickets sin necesidad de navegar por diferentes ventanas. En la opción del menú principal 'Tickets' podemos acceder a distintas ventanas donde los tickets, a su vez, también se encuentran categorizados. Por ejemplo, si accedemos a la opción 'Queue View' nos aparecen los tickets divididos por grupos (estos grupos también son modificables) como '2nd Level Support', '3rd Level Support', etc.

En definitiva, los tickets se encuentran categorizados de una manera muy útil.

5.- Nueva entrada a través de un email.

Tampoco permite la introducción de issues a través del e-mail.

6.- Posibilidad de obtener informes o estadísticas.

Los informes se lanzan a través de la ventana de búsqueda.

Buscar

Plantilla de búsqueda: - Crear Nueva

Estado: abierto
cerrado exitosamente
cerrado sin éxito
eliminado
mezclado

Agregar otro atributo: Ticket Número +

Formato de Resultados: Imprimir

Ejecutar Búsqueda

Seleccionando en '*Formato de Resultados*' imprimir exportamos la búsqueda en formato PDF, mientras que si seleccionamos CSV lo exportamos a formato Excel. Ésto unido a la eficiencia de las búsquedas hacen que los informes de OTRS sean útiles.

Diferente es el tema de las estadísticas. OTRS no ofrece la posibilidad de crear estadísticas o gráficas sobre información relacionada con los tickets, solo se limita a ofrecer la posibilidad de exportar los resultados de las búsquedas.

7.- Posibilidad de comentar o valorar una solución.

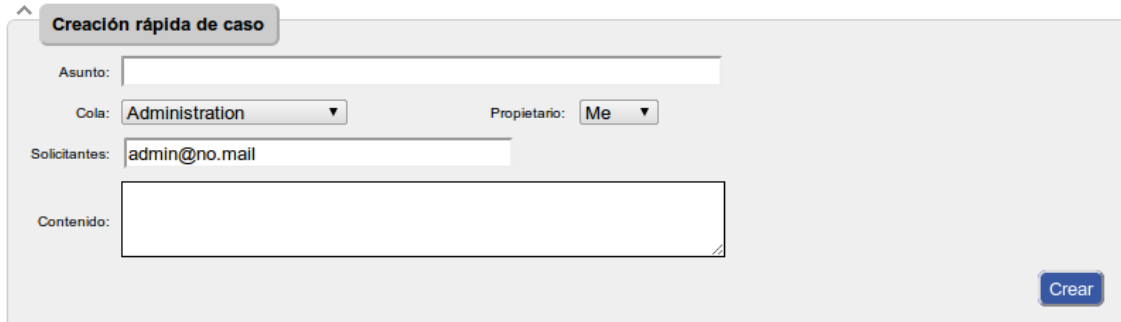
Existe la posibilidad de añadir notas a los tickets, calificándolas como interna o externa.

4.- REQUEST TRACKER

Request Tracker o RT [6], como se conoce, es un sistema bug-tracking, escrito en Perl, usado para el tratamiento y coordinación de tickets y gestionar las solicitudes entre una comunidad de usuarios.

1.- Fácilmente actualizable/introducción de datos.

De Request Tracker destacamos su herramienta de creación rápida de casos. La ventana de creación rápida es la siguiente:



Esta ventana está disponible desde el menú principal de la herramienta. Y es de gran ayuda a la hora de crear nuevos casos. Por esta funcionalidad podemos afirmar que la introducción de datos es rápida y eficiente.

2.- Eficacia en búsquedas.

RT tiene una búsqueda simple que no se caracteriza por el éxito de sus búsquedas. Sobre búsquedas avanzadas tiene una funcionalidad algo diferente a la vista en el resto de herramientas. Mientras que en las otras herramientas se rellenaban una serie de campos en esta se van añadiendo criterios pudiendo modificarlos una vez que se tenga la búsqueda avanzada definida.

Por esto podemos confirmar que sus búsquedas no son eficaces, ya que la búsqueda simple deja mucho que desear y la configuración de una búsqueda avanzada es muy pesada.

3.- Búsquedas personalizadas.

Es posible guardar las búsquedas avanzadas.

4.- Categorizado.

Existe la posibilidad de crear los casos en distintas 'Colas'. Lo que aporta todas las ventajas de la categorización de los issues. Es posible crear y configurar las colas. Estas son las opciones de la configuración:

- Nombre de la cola.
- Descripción de la cola.
- Subject Tag.
- Dirección de e-mail de respuesta.
- Dirección de comentario.
- Configuración de prioridades.
- Habilitar o deshabilitar.

5.- Nueva entrada a través de un email.

No es posible crear una entrada a través de un e-mail.

6.- Posibilidad de obtener informes o estadísticas.

Digamos que este es el punto más débil de RT. La funcionalidad de informes se limita solamente a tres opciones:

- Resuelto por propietario
- Resuelto en rango de fechas
- Creados en rango de fechas

No es posible crear ningún tipo de informe o estadística.

7.- Posibilidad de comentar o valorar una solución.

Existe la posibilidad de aprobar o no una solución pero no de comentarla o valorarla.

5.- FOOTPRINTS



Footprints es un sistema bug-tracking, basada en web, usado para el tratamiento y coordinación de tickets y gestionar las solicitudes entre una comunidad de usuarios.

1.- Fácilmente actualizable/introducción de datos.

La introducción o modificación de datos en Footprints no es una tarea rápida. Por una parte tenemos la ventaja de que se pueden configurar los campos obligatorios, por lo que si que podemos afirmar que la introducción de los datos en los tickets es rápida. El problema de este punto reside en la dificultad de seleccionar el ticket deseado si no sabemos el número del ticket o no tenemos ninguna búsqueda personalizada guardada, ya que la información en el menú principal de los tickets no es para nada completa y la interfaz no es nada clara.

Por estos dos motivos no podemos afirmar que el manejo de datos en Footprints sea rápido, pero tampoco que sea un punto débil de la herramienta.

2.- Eficacia en búsquedas.

Como en las anteriores herramientas Footprints cuenta con dos tipos de búsquedas, avanzadas y simples.

2.1.- Búsquedas Simples:

A través de este pequeño menú se puede realizar una búsqueda por título, palabra de la descripción o número de ticket. El punto fuerte de esta parte es que la búsqueda por palabra devuelve resultados muy exactos.

2.2.- Búsquedas Avanzadas:

Al igual que en el resto de herramientas es posible realizar una búsqueda avanzada. Poniendo como base la funcionalidad que te permite elegir que campos son obligatorios y cuales no, si este número de campos es pequeño estás búsquedas avanzadas ganan en agilidad y por lo tanto en eficacia.

Podemos asegurar que Footprints cuenta con unas búsquedas más que optimas.

3.- Búsquedas personalizadas.

Es posible guardas las búsquedas avanzadas para ejecutarlas de una manera más rápida en la herramienta. Al hecho de que estás búsquedas son optimas se une la facilidad de su uso.



Con este combo se seleccionan las búsquedas personalizadas y se ejecutan al instante.

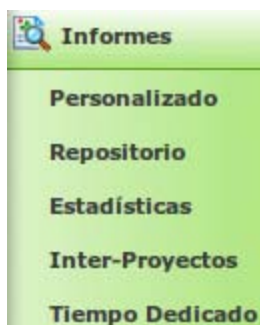
4.- Categorizado.

El punto más débil de esta herramienta es que no cuenta con una categorización de los tickets.

5.- Nueva entrada a través de un email.

A diferencia del resto de herramientas Footprints permite la creación de tickets a través del e-mail. Al enviar un e-mail a la dirección de Soporte de Openbravo automáticamente se crea un nuevo ticket. El débil de esta funcionalidad es que no es posible introducir información en el campo 'Company' de los tickets, el más importante de todos. Por esto Footprints tiene esta funcionalidad aunque no sea para nada útil.

6.- Posibilidad de obtener informes o estadísticas.



Estas son las opciones que ofrece la herramientas respecto al punto de informes y estadísticas.

Paso 1: Estilo

☒ Columnas
☐ Ajustado
☐ Operaciones/Gráficos
☐ Exportar

☐ **Combinado** Los Informes de estilo Columnas y Ajustado también pueden contener Operaciones. Marcar esta casilla para incluir Operaciones.

OK

6.1.- Personalizado:

Con esta opción puedes crear un informe a tu gusto. Primero eliges el estilo del informe pudiendo elegir entre :

- Columnas: Cada Ticket es una línea de una tabla
- Ajustado: Una lista de Tickets. Permite saltos de línea; óptimo para informes que incluyan muchos campos.
- Operaciones/Gráficos: Muestra sumas, medias y contadores.
- Exportar: Exportar datos a un archivo de texto en su equipo.

Tras elegir el estilo se debe realizar la consulta del informe a través del mismo menú que las búsquedas avanzadas.

Este es el punto más fuerte de la herramienta.

7.- Posibilidad de comentar o valorar una solución.

En este apartado podemos hablar de dos puntos diferentes en la herramienta, el comentario de los tickets y la valoración de soluciones de la Base de Conocimientos.

7.1.- Comentario de las incidencias:

Al modificar un ticket es posible, a parte de añadir información o modificarla en el campo 'descripción', realizar una comunicación interna. A través del campo 'Internal Communication' es posible realizar comentarios sobre las incidencias que solo serán visibles por el equipo de soporte.


7.2.- Valoración de Soluciones:

Cuando entras a consultar una solución de la Base de Conocimientos nos encontramos el siguiente apartado:

¿Le ha sido útil esta Solution?

☐ Sí ☐ No

Esta pregunta añade un punto en caso afirmativo a la valoración (campo popularidad) de la Solución. De esta manera cuando realicemos búsquedas de soluciones podemos elegir la más valorada por los usuarios.

Popularidad 
3
2

Resumimos el estudio de las herramientas en la siguiente tabla:

Características ----- Herramientas	Facilidad de introducción datos	Búsquedas eficaces y eficientes	Búsquedas personalizadas	Categorizado	Nueva entrada por e-mail	Posibilidad informes y estadísticas	Posibilidad de valorar solución
MANTIS	0.5	0	1	0.5	0	1	0.5
BUGZILLA	0.5	0	0	0	0	1	0.5
OTRS	1	1	1	1	0	0	0
REQUEST TRACKER	1	0	1	1	0	0	0.5
FOOTPRINTS	0.5	1	1	0	0.5	1	1

El resultado sería el siguiente:

- 1.- Footprints: 5 puntos
- 2.- OTRS: 4 puntos
- 3.- Request Tracker: 3.5 puntos
- 4.- Mantis: 3.5 puntos
- 5.- Bugzilla: 2 puntos

5.- CONCLUSIÓN

5.1.- ELECCIÓN DE LA HERRAMIENTA

Tras los pasos realizados llegamos al punto de elegir una de las herramientas estudiadas para su utilización en el equipo de Soporte.

La decisión que se toma se basa en diferentes aspectos:

1.- Coste Económico:

Es importante minimizar lo máximo posible el coste económico de la implantación de esta herramienta. Desde este punto de vista el coste no varía mucho de una herramienta a otra. Ya que en muchas es necesario tener un número de licencias, un contrato de soporte para posibles problemas, etc.

2.- Coste Temporal:

La implantación de la herramienta debe afectar lo menos posible al trabajo diario del equipo de Soporte y el servicio debería estar activo lo antes posible. En este punto se tiene en cuenta que un cambio de herramienta equivaldría a una migración de datos con todos los costes que esto implica.

3.- Calidad de la herramienta:

Teniendo en cuenta el estudio de las herramientas basándonos en las necesidades transmitidas del equipo de soporte también se analiza otro tipo de calidad de la herramienta (velocidad de la herramienta, interfaz amigable, etc.). Las herramientas estudiadas no varían mucho en estos aspectos, por lo que no tratamos este punto como importante en la decisión de la elección.

4.- Resultado a corto-medio plazo:

Cuando se empezó el proyecto se quería tener resultados lo antes posibles. Se quería que las ventajas de utilizar la Base de Conocimientos de una herramientas se vieran a corto-medio plazo en el trabajo del equipo de Soporte. Este resultado no se basa en la dificultad de la utilización de la base de conocimientos de cada herramientas, sino en el tiempo que se debiera dedicar en la implantación de la herramientas hasta el punto de estar operativa al 100%.

Teniendo en cuenta estos puntos se decidió mantener la herramienta actual en el equipo de soporte, Footprints. Analicemos el por qué:

1.- Coste Económico:

Como comentaba en el apartado anterior la diferencia de coste económico entre las herramientas no es lo suficientemente grande como para eliminar una respecto a las demás. El gasto que se tiene actualmente de la utilización de Footprints no disminuiría de manera excesiva con el resto de herramientas teniendo en cuenta que actualmente Openbravo cuenta con cuatro usuarios fijos y dos usuarios concurrentes más el soporte que proporciona Footprints.

2.- Coste Temporal:

Tener la herramienta funcionando actualmente en el equipo de Soporte aporta la mayor ventaja a Footprints respecto al resto. Evitaríamos la migración de datos, la necesidad de cursos o tiempo en la utilización de la herramienta para conocer toda su funcionalidad, evitaríamos también el impacto producido a los Partners y clientes el cambio de herramienta.

3.- Calidad de la herramienta:

En el estudio realizado en este proyecto Footprints sale como la herramienta más valorada para el equipo de soporte. Esto, unido a que el resto de diferencias entre herramientas no es relevante, hace que la valoración sobre Footprints sea mayor.

4.- Resultado a corto-medio plazo:

Partiendo de que el coste temporal es muchísimo menor el resultado a corto-medio plazo esta totalmente asegurado. En este punto solo faltaría realizar el estudio de la base de conocimientos y su modo de utilización para poder ponerla en funcionamiento.

5.2.- BASE DE CONOCIMIENTO DE FOOTPRINTS

Actualmente Footprints cuenta con su propia Base de Conocimiento, a continuación pasamos a realizar una descripción de esta parte de la aplicación.

Comenzamos hablando de la barra de menú desde la cual podemos acceder a las diferentes partes de la aplicación. Antes de centrarnos en la Base de Conocimiento pasamos a realizar una breve descripción de cada parte de este menú.

– Inicio del Proyecto: con este botón accedemos a nuestro tablero de mandos, el lugar donde aparece la lista de las incidencias de cada usuario. Esta parte es configurable con el fin de mostrar los datos deseados de cada incidencia. Desde la criticidad hasta los tiempos de resolución o de respuesta (*SLA*) pasando por el estado de la misma.

– Ticket: de esta manera accedemos a la creación de un nuevo ticket.

– Contactos: Muestra la lista de contactos. La compañía a la que pertenece, el tipo de soporte que tiene contratado, el ID y el nombre del usuario.

– Buscar: Las búsquedas de incidencias en Footprints pueden ser rápidas, con una simple búsqueda sobre el título, la descripción o su número; también existen búsquedas avanzadas, donde podemos introducir más datos con el fin de realizar una búsqueda más acotada y eficaz.

– Informes: la aplicación permite realizar informes detallados de todo lo relacionado con la gestión de incidencias.

– Panel de Control: esta acción nos muestra las estadísticas de cada día hasta ese momento. Estas estadísticas hacen referencia al número de incidencias activas y no activas tanto para el usuario como para el equipo de soporte.

– Base de Conocimiento: más tarde pasaremos a realizar una descripción más exhaustiva de la base de conocimiento.

– Conexiones: esta funcionalidad te permite abrir una comunicación tipo *chat* con los clientes.



- Preferencias: desde aquí realizaremos distintas modificaciones sobre la aplicación. Desde cambios en los datos generales del usuario hasta modificaciones sobre qué información queremos que se muestre en el tablero de mandos.
- Calendario: el calendario nos muestra una agenda con la que poder llevar una mejor organización.
- Ayuda: aquí podemos encontrar toda la información relacionada con Footprints.
- Salir: de esta forma salimos de la aplicación.

Una vez descrito el menú con el que nos encontramos vamos a introducirnos, con más detalle, en la parte que nos interesa, la Base de Conocimiento.

Podemos realizar sobre la Base de Conocimiento dos principales acciones:

- 1.- La introducción de una Solución
- 2.- Búsqueda de una Solución.

1.- La introducción de una Solución.

La introducción de una Solución tiene una serie de etapas. Todas ellas son importantes para que esta Solución sea de calidad.

- Etapa 1: Resolución de la incidencia:

Todo comienza en el momento que una incidencia es asignada a un miembro del equipo de soporte. La calidad de la Solución agregada a la Base de Conocimiento depende directamente de la calidad de la resolución de la incidencia ya que la descripción ésta será la que de cuerpo a la Solución. Por todo ello si la calidad de la respuesta de la resolución de la incidencia es buena, la calidad de la Solución relacionada también lo será.

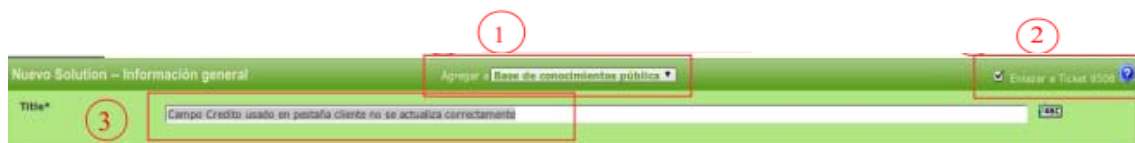
- Etapa 2: Agregar la Solución:

Una vez resuelta la incidencia es hora de pasar a introducir la Solución en nuestra Base de Conocimiento. En la parte superior de cada incidencia nos encontramos con un pequeño menú donde se indican las acciones que podemos realizar sobre el ticket.



Entre todas estas acciones tenemos la de 'Agregar a base de conocimientos'. Esta acción nos crea la Solución y nos la abre en modo edición. También existe la posibilidad de crear una solución sin tener un ticket asociado. Para ello bastaría con seleccionar la opción 'Agregar Solución' en nuestro menú principal. Vamos a describir los pasos correctos para que nuestra Solución sea de calidad.

1.- Información general



1.1- Dentro de la base de conocimientos existen diferentes tipos. Por defecto nos encontramos con dos:

- La base de conocimiento Pública: es accesible tanto para clientes como para el equipo de soporte.
- La base de conocimiento Privada: es accesible únicamente para el equipo de soporte.

En esta parte deberemos seleccionar a cuál de las dos bases de conocimiento queremos que pertenezca nuestra solución.

1.2- Tenemos la posibilidad, en el caso de que la solución se haya creado a partir de un ticket, de no relacionar nuestra solución con dicho ticket. Para ello bastaría con deschequear esta opción.

1.3- El título es la parte más importante de la Información General. Es fundamental que este título sea descriptivo y que haga referencia de forma concreta al tema de la solución, ya que las búsquedas rápidas se basan en este campo.

La aplicación crea la solución con el mismo título que la incidencia. Las incidencias, en la mayoría de los casos, son creadas por los Partners o Clientes, por lo que en general, los títulos no describen de manera correcta la posible solución. Por todo esto, a la hora de crear la solución se deberá modificar este título.

The screenshot shows a web form titled 'Información de Issue'. It contains several dropdown menus and text input fields. Numbered callouts (1-11) point to the following fields:

- 1: Task (dropdown, value: Developing)
- 2: Category* (dropdown, value: Regular)
- 3: Project (empty text input)
- 4: Application (dropdown, value: Openbravo)
- 5: Application Area (dropdown, value: MDO - Report)
- 6: Element (dropdown, value: PDF)
- 7: Openbravo Module (dropdown, value: Warehouse Management)
- 8: SLACumplido (dropdown, value: SI)
- 9: ResponseSuperado (dropdown, value: SI)
- 10: Time dedicated* (text input, value: 00:25)
- 11: OBProject* (dropdown, value: sup001)

2.- Información del Issue:

2.1.- *Task* (Tarea): En este combo seleccionamos el tipo de tarea o cometido relacionado con la solución. Podemos incluir la solución dentro de los siguientes grupos:

- *Sin opción*: de esta forma no añadimos información en este campo.
- *Buying a Support Package*
- *Commercial*
- *Configuring*
- *Developing*
- *Installing*
- *Localizing*
- *Maintaining*
- *Reporting a Bug*
- *Specialized Consulting*
- *Using as a final user*
- *Other...*

2.2.- *Category*: la categoría hace referencia a la categoría de la incidencia. Ésta puede ser *Regular*, en los casos normales, o *Extended*, en los casos en los que la incidencia esté relacionada con algún desarrollo.

2.3 – *Project*: Podemos definir un proyecto para cada solución. Este campo es independiente del campo *OB Project*, el cual hace referencia a los proyectos de Openbravo.

2.4- *Application*: seleccionamos la aplicación a la que hace referencia la solución. Las

aplicaciones son las siguientes:

- *Openbravo*
- *Tomcat*
- *Database*
- *Java*
- *Operating System*
- *Hardware System*
- *Hardware Related*

2.5 – *Application Area*: en el caso de que la aplicación seleccionada sea Openbravo, aparece otro combo, en el cual elegiremos el área de la aplicación a la que la solución hace referencia.

Existen las siguientes áreas:

- *Manual Reporting*
- *Manual Transaction*
- *Menu*
- *Windows*
- *Forms*
- *Report*

2.6 – *Element*: en los casos en los que el área seleccionada sea *Manual Reporting*, *Report* o *process* surge este combo donde deberemos seleccionar el elemento del área al que la solución hace referencia. Estos son los elementos que podemos elegir.

- *Manual Reporting*:
 - *Callout*
 - *Validation Rule*
 - *References*
 - *Auxiliar Inputs*
 - *Table & Column*
 - *Window, Tab and Field*
 - *Message*
- *Report*:

- *HTML*
- *Excel*
- *PDF*
- *SubReports & Links*
- *Dimensional*
- *Ajax*
- *Process:*
 - *Background*
 - *PL*
 - *Forms*
 - *Java*
 - *Action Buttons*

2.7.- *Openbravo Module*: en este combo seleccionamos el módulo de Openbravo al que hace referencia la solución. Las opciones disponibles son:

- *Procurement* (Compras)
- *Warehouse* (Almacenes)
- *Project and Services* (Proyectos y servicios)
- *Product* (Producción)
- *Sales* (Ventas)
- *CRM*
- *Financial & Economics* (Finanzas)
- *Business Intelligence*
- *Master Data* (Datos Mestros)

2.8.- *SLA Cumplido*: este campo informa si el SLA se ha cumplido o no.

2.9.- *Response Superado*: este es un campo no editable donde se muestra si el tiempo de respuesta se ha superado o no.

2.10.- *Time dedicated*: muestra el tiempo dedicado a la incidencia, es modificable con el fin de añadir el tiempo que se ha dedicado a la creación de la Solución. Este campo se rellena automáticamente.

2.11.- *OB Project*: en este combo seleccionamos el proyecto de Openbravo que hace referencia la Solución. Este campo se rellena automáticamente.

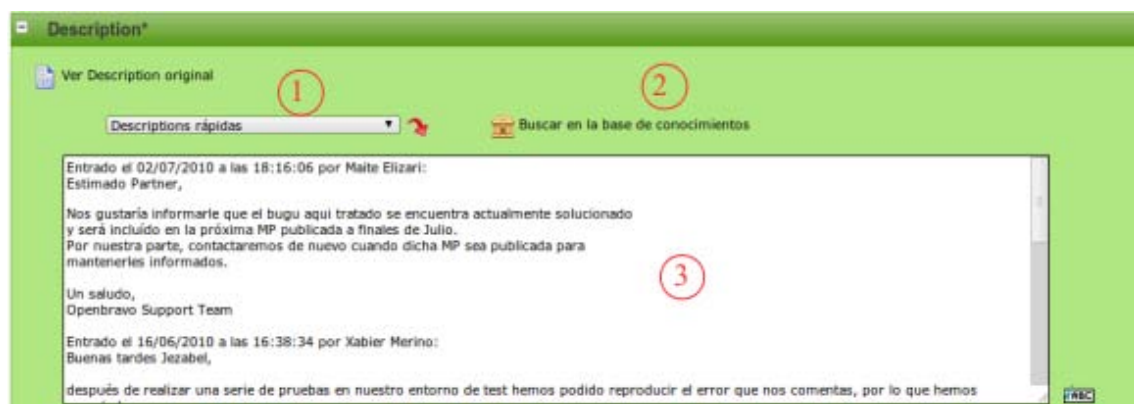
3.- Bug Details:



3.1.- *Issue in Mantis*: este campo muestra la dirección del bug perteneciente a la incidencia en nuestro repositorio de *Issues*. Este campo se rellena automáticamente cogiendo el contenido del campo con el mismo nombre de la incidencia.

3.2.- *Consulting Project*: en este campo se selecciona el proyecto de consultoría al que pertenece la incidencia. También es un campo que se rellena de manera automática.

4.- Description:



4.1.- *Descripciones rápidas*: existen una serie de comentarios estándar que el usuario puede usar para realizar la contestación de alguna incidencia de una manera más

rápida. Estas descripciones tienen sentido en las incidencias, no mucho en las soluciones.

4.2.- Buscar en la base de conocimientos: existe la posibilidad de completar la nueva solución con información de otras soluciones.

4.3.- Este campo la base de la Solución, y es el lugar al cual más tiempo se le debe dedicar para conseguir una Solución de calidad. En los casos en los que se agregue una solución a la Base de Conocimiento este apartado aparecerá en blanco, en cambio, si la solución se crea a partir de una incidencia, en este campo se mostrará todas las comunicaciones entre Openbravo y Partner.

Existen dos puntos importantes que se tendrán en cuenta a la hora de agregar la Solución. El primero de ellos hace referencia cual de las Bases de Conocimiento se va a agregar, si a la privada o a la pública. Hay que tener en cuenta que en la comunicación es posible que salgan datos confidenciales o de importancia, tales como e-mails, datos de cuentas FTP de los partners, IP's y usuario y contraseñas. Por ello, en los casos en los que la solución se agregue a la base de conocimiento pública se tendrán que borrar este tipo de datos.

El siguiente punto se centra en la organización de la información. Para que una solución sea de calidad es importante que sea clara y organizada, por esto se deberá dividir este apartado en dos temas, 'problema' y 'solución'.

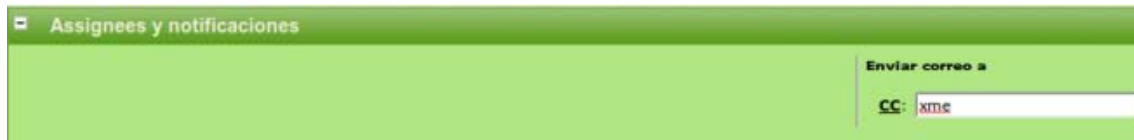
5.- Adjuntos:



Con esta opción se permite adjuntar ficheros que ayuden a aportar calidad a la solución. Si la incidencia desde donde se crea la Solución tiene archivos adjuntos, éstos aparecerán en la Solución. Aquí es importante recordar si nuestra Solución se incluirá en

la Base de Conocimiento pública con el fin de eliminar cualquier tipo de dato confidencial.

6.- Assignees y notificaciones:



En este apartado se incluyen los destinatarios a los que llegará una notificación vía e-mail de la creación de la solución. Por defecto se muestran las direcciones del contacto del partner y del responsable de soporte.

Se deberá modificar para que la notificación llegue al grupo de soporte o a otra gente interesada.

De esta manera habremos introducido una Solución de calidad que enriquezca nuestra base de conocimiento y que ayude a que las futuras búsquedas sean eficaces.

2.- Búsquedas.

Una vez descrito los pasos a seguir para una correcta introducción de una Solución vamos a pasar a explicar la forma de realizar una búsqueda.

Podemos diferenciar dos tipos diferentes de búsquedas:

- Búsquedas a través de Footprints.
- Búsquedas a través de Thunderbird.

1.- Búsquedas a través de FootPrints:

En el menú principal tenemos la opción 'Buscar' en la Base de Conocimiento. En las búsquedas de FootPrints, a su vez, podemos encontrar otros dos tipos de búsquedas, las búsquedas sencillas y las búsquedas avanzadas. Vamos a analizar cada una de ellas.



1.1.- Búsquedas sencillas:

Accedemos a las búsquedas sencillas a través del botón 'Buscar' de la opción de Base de Conocimiento de nuestro menú principal.



1.- En la opción 'Base de conocimientos' elegimos el tipo de base en la que queremos realizar la búsqueda, Privada, Pública o ambas.

2.- Cuando elegimos la Base de conocimiento pública tenemos la opción de acotar más la búsqueda.

Por ejemplo, por proyectos.

3.- Podemos realizar la búsqueda a través de palabras clave. En este punto podemos ver la importancia de que las soluciones tengan títulos descriptivos ya que sino la búsqueda rápida sería muy poco eficaz.

4.- 'Cambiar a modo avanzado'. De esta forma accedemos a las Búsquedas avanzadas.

1.2.- Búsquedas avanzadas:

Este modo es muy completo. Es posible elegir todo tipo de características para acotar la búsqueda lo máximo posible. Vamos a realizar una descripción detallada de todas las opciones. Hay que tener en cuenta que para que estas búsquedas tengan éxito las Soluciones deberán haber sido agregadas con el mayor número de datos posible.

Al igual que las búsquedas sencillas, se permite elegir si queremos realizar las búsquedas sobre la base de conocimiento privada, pública o sobre las dos.



1.- *Title*: introducimos palabras clave que se buscarán en los títulos.

2.- *Description*: introducimos palabras clave que se buscarán en las descripciones.

3.- *Palabra clave* (en cualquier campo): introducimos palabras clave que se buscarán tanto en el título, en la descripción, y en los archivos adjuntos.

4.- *Nombre de Archivo Adjunto*: introducimos palabras clave que se buscarán en los archivos adjuntos.

5.- *Fecha*: existen varias opciones en este apartado. Podemos elegir que la fecha sea la de la creación o la de la última modificación de la solución. También tenemos la opción de buscar sobre un rango de fechas:

5.1- *Exacta* (imagen superior): podemos buscar periodos anteriores a una fecha ('antes de'), posteriores ('después de'), o entre dos ('entre'), así como la fecha exacta ('en').

5.2- *Relativa*: con esta opción se permite elegir las soluciones creadas o modificadas en los últimos X (opción 'seleccione número') horas, días, semanas

o años.

5.3- *Periodo*: en esta opción señalamos un periodo concreto. Por ejemplo: hoy, ayer, anteayer, lunes de la semana pasada, martes de la semana pasada, esta semana, la semana pasada, este mes o el mes pasado.

6.-

Antigüedad

: elegimos la antigüedad de la solución. Elegimos si queremos que la antigüedad sea mayor que, menor que, igual a, mayor o igual a, menor o igual a, distinto de unos días y unas horas.

7.- *Número ticket*: con esta opción buscamos las soluciones que existan en un rango de incidencias o tickets.

Información del ticket: en este apartado se introducen las características que

Task	Buying a Support Package Commercial Configuring Developing Installing	Category	Regular Extended (sin datos)
Application	Openbravo Tomcat Database Java Operating System	Application Area	MVC - Manual Reporting MVC - Manual Transaction MDD - Menu MDD - Window MDD - Forms
Openbravo Module	Procurement Warehouse Management Project/Service Management Product Management Sales Management	Features	Ordering Supplies Materials Requirements Supplies Invoicing Project Proposals Managing Expenses
ResponseSuperado	SI NO (sin datos)	Time dedicated	
Issue in Mantis		Consulting project	BEROTZ999 BRASCUBA001 CIVERTICE002 COLA001 CONCORDIA001

Project	
Element	PL Forms Java Action Buttons (sin datos)
SLACumplido	SI NO (sin datos)
OBProject	ALIMARKET ARRASATE999 ARZOBISPADO999 CAFENASA999 Cuñado

tiene nuestra
solución a buscar.

8.- Existe la posibilidad de ordenar los datos de salida en función de diferentes campos.

Opciones Avanzadas:

Ordenar resultados por 8

Campo:	Popularidad ▼	Orden:	descendente ▼
Campo:	nada ▼	Orden:	ascendente ▼
Campo:	nada ▼	Orden:	ascendente ▼

9

☐ Hacer la búsqueda por Title, Description, y campos de texto sensitiva a mayúsculas.

10 ☒ Y ☐ O Si se seleccionan múltiples condiciones, seleccionar un operador entre ellas.

correspondientes a la información de la solución, que acabamos de nombrar y los siguiente:

- *Popularidad*
- *Número de Ticket*
- *Remitente*
- *ID del usuario*
- *Fecha de envío*
- *Fecha de modificación*

9.- Podemos elegir que la búsqueda sea '*Case Sensitive*', es decir, que diferencie entre mayúsculas y minúsculas.

10.- En los casos que se elijan varias condiciones elegimos el operador entre ellas ('y', 'o').

Después de describir todas las opciones que tenemos en las búsquedas avanzadas vamos a hablar de la posibilidad de realizar búsquedas a través de Thunderbird (u otro clientes de correo)



Mozilla Thunderbird [7] es un cliente de correo electrónico de la Fundación Mozilla. El objetivo de Thunderbird es desarrollar un Mozilla más liviano y rápido

mediante la extracción y rediseño del gestor de correo de Mozilla oficial. Es multiplataforma, utiliza lenguaje de interfaz XUL y es software libre.

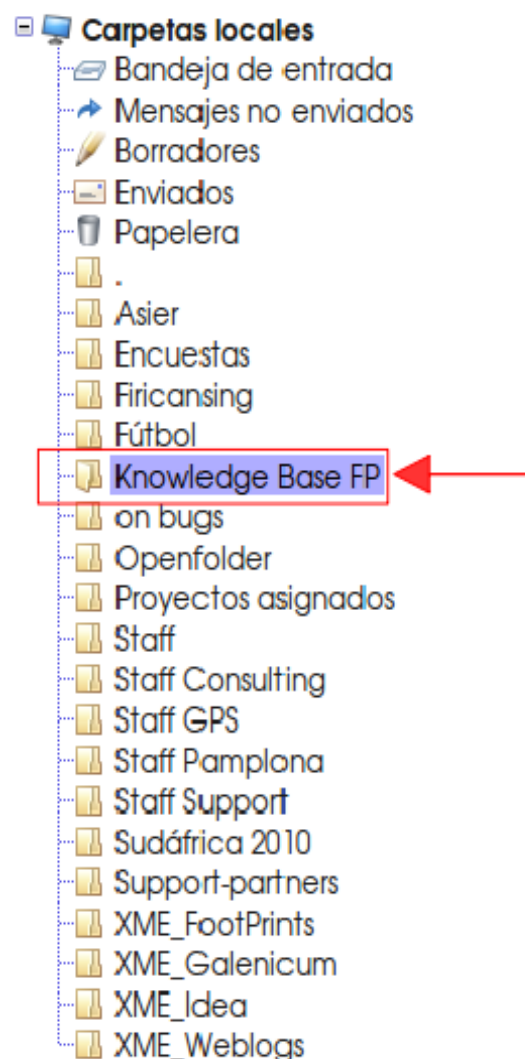
Como muchos clientes de correo electrónico, Thunderbird ofrece la posibilidad de filtrar los correos recibidos según una serie de criterios y de esta manera poder distribuirlos en diferentes carpetas.

Al igual que ocurre con las incidencias, FootPrints envía un e-mail a los destinatarios definidos cada vez que se crea o se modifica una solución.

Con estas dos características de nuestros dos programas pasamos a explicar la manera de organizar nuestro correo para poder realizar búsquedas sin necesidad de acceder a FootPrints.

1.- Crear una carpeta específica en Thunderbird:

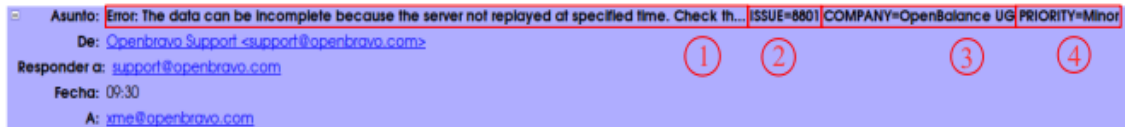
El primer paso es crear una carpeta donde guardar los e-mails recibidos en las creaciones y modificaciones de Soluciones.



2.- Crear el filtro adecuado.

Los e-mails recibidos desde FootPrints tienen diferentes Asuntos según la procedencia del mismo. Por ejemplo:

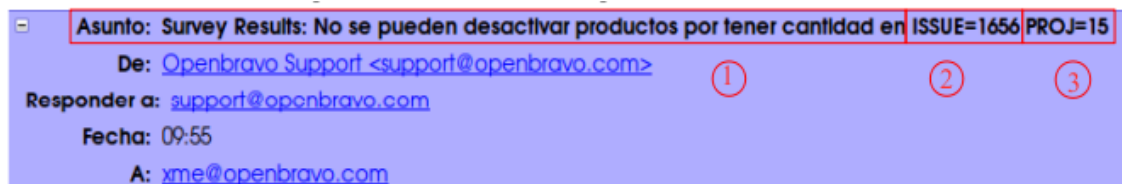
- E-mails recibidos a partir de una incidencia:



El asunto en estos casos se divide en:

- 1.- Título de la incidencia.
- 2.- Número de la incidencia.
- 3.- Nombre de la compañía.
- 4.- Criticidad de la incidencia.

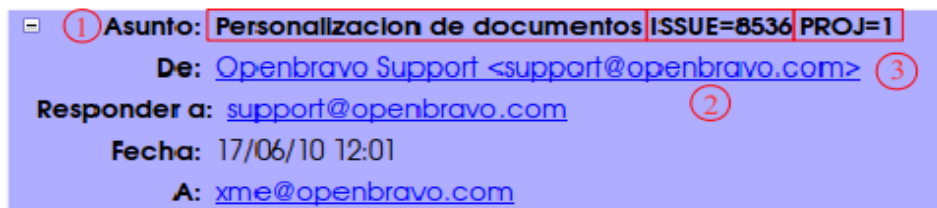
- E-mails recibidos a partir de encuestas de los partners:



El asunto en estos casos se divide en:

- 1.- Título de la encuesta.
- 2.- Número de la encuesta.
- 3.- Número del proyecto asociado a FootPrints.

- E-mails recibidos a partir de creaciones o modificaciones de Soluciones:

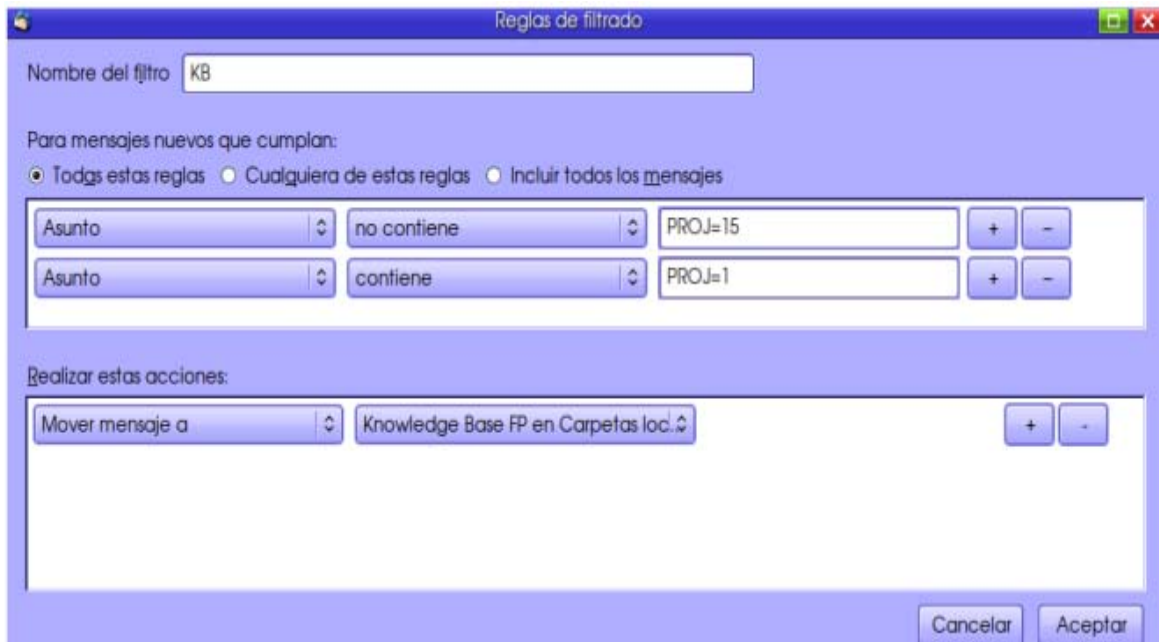


El asunto en estos casos se divide en:

- 1.- Título de la solución.
- 2.- Número de la solución.
- 3.- Número del proyecto asociado a FootPrints.

Viendo estos ejemplos podemos concluir que 'PROJ=1' es la parte del asunto que diferenciaa los e-mails recibidos relacionados con las soluciones de los relacionados con las incidencias. El asunto de los relacionados con las soluciones no contiene 'PROJ=15', de esta manera los diferenciamos de los relacionados con las encuestas.

Pasamos a crear el nuevo filtro. Quedaría de la siguiente manera:



Se definen los criterios:

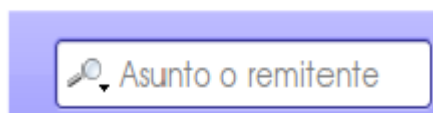
Asunto no contiene PROJ=15 y contiene PROJ=1.

Y las acciones a realizar:

Mover mensaje a Knowledge Base FP (la carpeta creada en el paso 1).

De esta manera los e-mails que recibamos por cada creación o modificación de una solución se guardaran en esta carpeta.

3.- Búsqueda.



Thunderbird ofrece la posibilidad de realizar una búsqueda rápida. Ésta búsqueda se realizar sobre los asuntos o los remitentes, por este motivo destacábamos anteriormente la importancia del título de las Soluciones.

De esta manera seremos capaces de realizar búsquedas rápidas sin necesidad de entrar en la aplicación, evitando de esta manera el coste temporal que supone.

5.3.- LÍNEAS FUTURAS

A partir de este proyecto Openbravo quiere sacar el máximo partido posible a las ventajas que ofrece la utilización de una base de conocimientos. Para ellos se han pensado una serie de propuestas para el futuro. Podemos dividir estas propuestas en tres tipos: a corto plazo, a medio plazo y a largo plazo.

Lanzada ya la base de conocimientos en el equipo de Soporte, el siguiente paso a corto plazo sería incluir a Partners y Clientes en la utilización de esta herramienta. Tanto unos como otros utilizan actualmente la herramienta Footprints para hacer llegar al equipo las diferentes incidencias. Utilizando la base de datos pública de la base de conocimiento se podría ofrecer la posibilidad a Partners y Clientes de realizar consultas de las dudas más comunes o de los bugs ya resueltos en la aplicación, ahorrándose así la introducción de una nueva incidencia. De esta manera ellos obtendrían un ahorro de tiempo y de crédito de soporte y la carga de trabajo del equipo sería menor.

Otro paso a tomar por parte de la empresa sería la de incluir a otros equipos que conforman la empresa en el uso de la herramienta. Por ejemplo, el equipo de Consultoría podría acceder a la base de conocimiento para conocer de primera mano las dudas más generales de los Partners y Clientes.

Respecto a las posibilidades a largo plazo la propuesta más ambiciosa de Openbravo es desarrollar una herramienta propia tanto para la gestión de incidencias en el equipo de Soporte como para la gestión de soluciones. Esta herramienta sería implantada en la misma aplicación Openbravo. El gran problema de este objetivo es el tiempo y los recursos. Un desarrollo de esta complejidad costaría o bien mucho tiempo o bien muchos recursos.

6.- GLOSARIO

6.1.- *Enterprise Resource Planning* (ERP):

La Planificación de Recursos Empresariales, o simplemente ERP, es un conjunto de sistemas de información gerencial que permite la integración de ciertas operaciones de una empresa, especialmente las que tienen que ver con la producción, la logística, el inventario, los envíos y la contabilidad. De esta manera, a través del software ERP, en vez de estar los programas trabajando de manera independiente uno del otro y sin tener conexión entre sí, trabajan de una forma integrada que permite la interconexión entre ellos.

La integración de todos estos datos en una base de datos centralizada permite la optimización de los procesos y la obtención de la información de manera más rápida y precisa.

Los ERP cuentan con más particularidades que los diferencian de otros *softwares* de gestión, y que son la división interna de módulos, lo que permite una mayor adaptabilidad a cada cliente.



6.2.- *Point of Sales*:

Punto de venta, proviene del ingles POS. Lugar o local en el que se atiende a clientes presencialmente, ya bien sea para venderles productos o prestar algún servicio. Estos locales pueden contar con sistemas informáticos especializados que ayudan en las tareas de gestión del Punto de Venta mediante una interfaz accesible para los vendedores que se denomina Terminal de Punto de Venta, los cuales realizan diversas operaciones durante todo el proceso de venta como cambios en el inventario o la creación de informes que ayuden en la gestión de negocio.



6.3.- Soporte Técnico:

El soporte técnico es un rango de servicios que proporcionan asistencia con el software o hardware de un ordenador, o algún otro dispositivo electrónico o mecánico. En general los servicios de soporte técnico tratan de ayudar al usuario a resolver determinados problemas con algún producto en vez de personalizar o entrenar.

La mayoría de compañías que venden hardware o software ofrecen soporte técnico de manera telefónica u *online*.

Cuando el soporte está debidamente organizado, se pueden dar varios niveles de soporte, donde el nivel 1 es el que está en contacto directo con el usuario y que soluciona las incidencias triviales, soporte nivel 2, daría al nivel que está por debajo y a este nivel llega información algo mas filtrada y así sucesivamente.

6.4.- Gestor de incidencias:

Un gestor de incidencias o sistema de seguimiento de errores (*Bug Tracking System*) es una aplicación informática diseñada para ayudar a asegurar la calidad de software y asistir a los programadores y otras personas involucradas en el desarrollo y uso de sistemas informáticos en el seguimiento de los defectos del software. Estos sistemas son usados intensivamente por empresas o instituciones que realicen desarrollos de software.

Muchos gestores de incidencias de software libre permiten que los usuarios directamente den de alta la incidencia detectada, en muchas empresas se usan de manera estrictamente interna.

Estos sistemas se integran frecuentemente con otras herramientas, como pueden ser correo electrónico, control de versiones y otras herramientas de gestión administrativa.

6.5.- Bussiness Partner:

Empresa que da lugar a una estrategia común con otra firma tras haber llegado a un acuerdo de carácter comercial, de implantación de nuevas tecnologías, de abastecimiento, etc.

6.6.- Acuerdo de Nivel de Servicio o *Service Level Agreement* (SLA):

Un acuerdo de nivel de servicio es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio. El SLA es una herramienta que ayuda a ambas partes a llegar a un consenso en términos de nivel de calidad de servicio, en aspectos tales como el tiempo de respuesta (*Response Time*), tiempo de resolución (*Resolution Time*), disponibilidad horaria, documentación disponible, personal asignado al servicio, etc.

6.7.- *Workaround*:

Un *workaround* es una solución rápida para solventar un posible bug mientras este se encuentra arreglando por el equipo de *Engineering*. Entre estas soluciones se encuentran desde pequeños cambios tanto en el código como en la base de datos, hasta cambios en la utilización de la aplicación para evitar el problema.

6.8.- Equipo QA (*Quality Assurance*):

El equipo de QA se encarga de testear la aplicación a través de tests manuales y automáticos.

6.9.- Parche:

En informática es una utilidad para la comparación de archivos que genera las diferencias entre los dos archivos o los cambios realizados en un archivo determinado comparándolo con una versión anterior del mismo archivo.

Esta utilidad genera un archivo . (fichero al que llamamos Parche) y éste es aplicado mediante la orden " para realizar el cambio de código.

Openbravo utiliza los parches para ofrecer soluciones a bugs antes de la salida de la siguiente versión de la aplicación. Estos parches se envían con el fin de cumplir el tiempo de resolución o por una petición expresa del Bussiness Partner o del cliente.

6.10.- Versión o *Maintenance Pack*:

Una versión de Openbravo (también conocida como o MP) es aquella en la que se incluyen mejoras y correcciones de bugs conseguidas desde la salida de la anterior versión.

6.11.- *Bug Tracking*:

Un sistema *Bug Tracking* es un sistema que permite notificar bugs de forma centralizada. Establece un flujo de trabajo, basado en estados, a partir de los bugs. Proporciona un sistema de notificación a través de emails, rss, etc. Es una herramienta muy importante de trabajo para el QA (*Quality Assurance*). Permite conseguir un *feedback* entre el usuario final y el grupo de desarrollo.

7.- BIBLIOGRAFÍA Y REFERENCIAS

7.1.- BIBLIOGRAFIA

- 1.- Wikipedia: <http://es.wikipedia.org/>
- 2.- Share: <http://eu-share.org/es/about-share.html>
- 3.- Michael Flanakin weblog: <http://www.michaelflanakin.com>
- 4.- Wikipedia Openbravo: <http://wiki.openbravo.com>
- 5.- Numara Footprints 7.5 'Getting Started Guide for Administrators'

7.2.- REFERENCIAS:

- 1.- <https://issues.openbravo.com/>
- 2.- <http://forge.openbravo.com/>
- 3.- <http://www.mantisbt.org/>
- 4.- <http://www.bugzilla.org/>
- 5.- <http://otrs.org/>
- 6.- <http://bestpractical.com/rt/>
- 7.- <http://www.mozillamessaging.com/es-ES/thunderbird/>